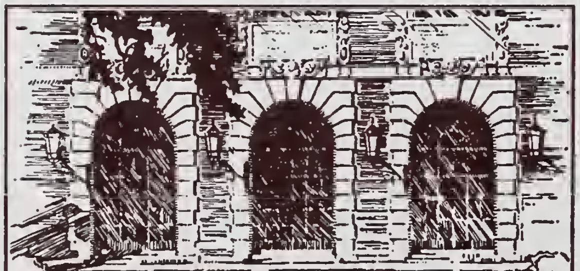


[redacted]

UNIVERSITY OF ILLINOIS
GRADUATE COLLEGE
DIGITAL COMPUTER LABORATORY

INTERNAL REPORT NO. 49

# ILLIAC ORDER CODE

By

J. P. Nash

October 12, 1953

# THE ORDER CODE

The Illiac is a binary computer in which the storage capacity of each register or memory location is 40 binary digits. The orders which the machine carries out are represented by numbers in the machine. The relation between the order types and the corresponding numbers is called a code, and the collection of all such numbers is called the order code of the machine. The order code is interpreted by the control circuits of the machine and completely determines what the machine does. The machine is designed so that any storage location in the memory may be used either for orders or for numbers, the only distinction being that the control must be instructed properly so that orders and numbers will be treated appropriately.

It is not necessary to use 40 binary digits to describe an order. The Illiac uses 20 digits and packs two orders (an order pair) into one location. These are the left hand order and the right hand order. Since many orders must refer to locations in the memory, each order contains an address. It is still called an address in those orders which do not refer to the memory.

The electrostatic memory of the Illiac has 1024 locations and because $1024 = 2^{10}$ we require 10 binary digits for the address. These are the rightmost 10 digits of the 20 digits assigned to an order.

Of the remaining 10 digits of each order, eight are used for the instruction or (function) and the other 2 are unused. The digit makeup of an order pair is then as in Figure 3.1.

| 8 DIGIT<br>INSTRUCTION | 10 DIGIT<br>ADDRESS | 8 DIGIT<br>INSTRUCTION | 10 DIGIT<br>ADDRESS |
|---|---|---|---|

LEFT HAND ORDER (DIGITS 0-19) | RIGHT HAND ORDER (DIGITS 20-39)

FIGURE 3.1 ORDER PAIR DIGIT MAKEUP

The instruction digits are 8 in number because of the convenience obtained by using a base 16 (sexadecimal) number system in which 4 binary digits may be represented by one sexadecimal digit. Thus each instruction may be coded as 2 sexadecimal digits. The symbols K, S, N, J, F, L are used for 10, 11, 12, 13, 14, 15.

As an example of an order pair, let us consider the following 40 binary digits.

1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0

When divided as in Figure 3.1, these digits look like this:

| | | | | | |
|---|---|---|---|---|---|
| 1111 0101 | 00 | 0000011101 | 0100 0000 | 00 | 0000000110 |
| INSTRUCTION | | L.H. ADDRESS | INSTRUCTION | | R.H. ADDRESS |

The left hand instruction is made up of the two 4-digit numbers 1111 0101 which are the sexadecimal digits L5. The left hand address is interpreted as an integer which may go from 0 to 1023 if we use decimal notation or from 0 to 3LL if we use sexadecimal notation.

In sexadecimal representation the left hand address is 1J which corresponds to the decimal number 29. Thus the left hand order is L501J where the 0 has been supplied so that all 20 binary digits (including the unused 2) are accounted for. (We could have set the unused digits to 1's and used N rather than 0 if we had wished.)

Similarly the right hand order is 40006, and we have, in sexadecimal notation, the order pair

L501J 40006.

This order pair says, "Transfer the contents of memory location 1J to the accumulator; store the accumulator contents at location 6 of the memory."

It is quite inconvenient to have to write addresses in sexadecimal form, and it is unnecessary if the machine contains a program which will take addresses written in decimal form and convert them to sexadecimal (i.e., binary) form for machine use. A program of this type is available for the Illiac. It is called the Decimal Order Input Routine.

The Illiac operates by using orders which have been stored in the memory according to a plan determined by the programmer. The program is begun with a particular order chosen by the programmer. Let us suppose that it is the left hand order at location 10. (We shall refer to addresses in decimal notation.) The control will put into $R_3$ the order pair from location 10. Then, until something is said to the contrary the control will follow a fixed pattern in executing orders. It will do the left hand order and then the right hand order in $R_3$. Then it will put the order pair at location 11 into $R_3$ and again do the left and right hand orders. It will continue to withdraw and execute order pairs from successive memory locations until one of two things occurs:

(a) One of the orders brought out says "stop",

(b) One of the orders brought out says "change the sequencing".

The second kind of order, a control transfer orde‸ permits the programmer to change the sequencing of orders and provides the flexibility required for iterative processes. It works in the following way. Let us suppose that after the machine has executed the left hand order at location 17 the programmer wishes to move to a sequence of orders beginning, say, with the right hand order at location 35. Then the right hand order at location 17 will say "Transfer Control to R.H. order at 35." The execution of this order will consist of arranging that the next order pair is brought to $R_3$ from location 35 and that the left hand order is skipped. Having done the right hand order in $R_3$, the control brings out the order pair from location 36 and proceeds in the usual way.

There are two kinds of control transfer orders, conditional transfers and unconditional transfers. We have just described the unconditional transfer. The conditional transfer is the same as the unconditional transfer provided that the number in the accumulator is not negative. Otherwise it does nothing. In the example we have just used, if the right hand order at location 17 had been conditional, then if the accumulator had held zero or a positive number the next order executed would have been the right hand order at location 35. But if the accumulator had held a negative number the next order executed would have been the left hand order at location 18.

There are four control transfer orders, the right and left unconditional and conditional orders. These have been further combined with a stop or not stop order, giving eight orders in all. If the programmer chooses one of the four "stop and transfer control orders" the machine will stop before transferring control and must be started again with a panel switch. The switch can be set so that these stop orders are ignored,

which of course makes these stop orders into ordinary control transfer orders. This is often useful in programming, and when it is done the machine can still be stopped with another kind of stop order which cannot be ignored.

The number of orders which the Illiac can execute is quite large (more than 100), but not all of them are of general interest. What we shall do here is discuss the various types of orders, giving the variants of each. Following this discussion will be a list of those Illiac orders which are of most use. This list is adequate for the coding of any problem and beginners are urged to confine themselves to it. Experienced coders will find uses for other variants.

The two sexadecimal instruction digits of an order give the order type and the order variant. We shall refer to them as the T- and V- digits, respectively. In the example L5 cited earlier the T- digit is L and denotes addition. The V-digit is 5 and chooses one of the variants of the addition order.

The order types are given in Table 3.1.

It will be seen from Table 3.2 that except for the 0, 1, 8, 9, K and S orders the address of an order always refers to the memory.

Let us now consider the variants obtained by changing the V- digit. The sexadecimal V-digit is made up of the four binary digits V8, V4, V2 and V1. There are 16 possible combinations, giving V-digits from 0 to L, but not all are used. The results obtained with different V-digits are as follows:

| T-Digit | Order Type |
|---|---|
| 0 | Left Shift |
| 1 | Right Shift |
| 2 | Unconditional Transfer |
| 3 | Conditional Transfer |
| 4 | Store from A Register |
| 5 | Memory to Q Register |
| 6 | Divide |
| 7 | Multiply |
| 8 | Input or Output |
| 9 | Special Input or Output |
| K | Increment Add from Q |
| S | Add from Q Register |
| N | Not Used |
| J | Extract |
| F | Increment Add |
| L | Add |

Table 3.1 ORDER TYPES

The meaning of the address digits of the different order types is given in Table 3.2.

| ORDER TYPE | ADDRESS SIGNIFICANCE |
|---|---|
| 0,1 | Number of shifts (cannot exceed 63) |
| 2,3 | Memory location from which next order pair will come |
| 4 | Memory location at which storage will occur |
| 5 | Memory location from which word is brought to Q register. |
| 6 | Memory location of divisor |
| 7 | Memory location of multiplicand |
| 80, 82 | Number of binary digits to be input or output |
| 92 | Character to be punched and number of punchings |
| K, S | Address not used |
| J | Same as for 5 order |
| F | Memory location of addend |
| L | Memory location of addend |

Table 3.2 MEANING OF ADDRESS DIGITS

(a) The Digit V1. If V1 = 1, A will always be cleared to zero at the beginning of any order. If V1 = 0, A will not be cleared. Thus an odd V-digit means clear A.

(b) The Digit V8. If V1 = 1 and V8 = 1, the quantity 1/2 will be put in A at the beginning of any order. This is how rounded multiplication is carried out. If V8 = 1 and V1 = 0, the Illiac will hang up.

(c) The Digit V2. This digit affects all orders except those of types 0, 1, 5 and J. See Table 3.3.

(d) The Digit V4. This digit affects all orders except those of types 0, 1, 5 and J. See Table 3.3.

The following notation is convenient for a more detailed description of the orders:

A = accumulator register

Q = multiplier-quotient register

AQ = the double register formed from A and Q, excluding $q_o$

N(R) = contents of register R

N(n) = contents of memory location n

$a_o$ = sign digit of A

$q_o$ = sign digit of Q

If no ambiguity is possible, the symbols A and Q will also be used to denote the contents of A and of Q.

| ORDER TYPE | V4 | EFFECT OF V4 | V2 | EFFECT OF V2 |
|---|---|---|---|---|
| 0,1<br>Shift | -- | No effect | -- | No effect |
| 2,3 | 0 | Transfer to right hand order. | 0 | If panel switch is set to OBEY, machine will stop before transferring control. |
| Transfer | 1 | Transfer to left hand order. | 1 | No effect |
| 4 | -- | No effect | 0 | Store full word |
| Store | 0<br>1 | Right hand address<br>Left hand address | 1 | Store address digits |
| 5,J | | | | |
| Memory to $R_2$ | -- | No effect | -- | No effect |
| 6 | | | | |
| Divide | 1 | No choice | 1 | No choice |
| 8 | 0 | Tape | 0 | Input |
| Drum, Tape | 1 | Drum | 1 | Output |
| 9 | | | | |
| Letter Output | 0 | Tape | 1 | Output |
| 7, K, S, F, L | 0 | Subtract | 0 | Use number in memory or Q |
| Multiply and Add | 1 | Add | 1 | Use absolute value of number in memory or Q |

TABLE 3.3 EFFECTS OF V4 and V2

On the following pages there are detailed descriptions of the results obtained by changing the V-digit of the 15 order types used in Illiac. After the detailed descriptions is an abbreviated list of orders. The orders in the abbreviated list have been underlined in the detailed descriptions.

| | | |
|---|---|---|
| OV n | Left Shift of AQ<br>Final Stop | 3 Orders<br>1 Order |

If n = 0, the machine will stop. If not, repeat n times the operation which replaces the contents

$a_0, a_1, a_2, \dots, a_{38}, a_{39}$ of A

and

$q_0, q_1, q_2, \dots, q_{38}, q_{39}$ of Q

by

$a_1, a_2, a_3, \dots, a_{39}, q_1$

and

$q_0, q_2, q_3, \dots, q_{39}, 0.$

The number n will be interpreted modulo 64.

## Variants

| | |
|---|---|
| 0, 2, 4, 6 | Order as described above. |
| 1, 3, 5, 7 | Clear A and then execute as described. |
| 8, K, N | Illiac will hang up. Avoid these. |
| 9, S, J, L | Clear A, insert 1/2 in A, and then execute order as described above. |
| F | Final stop. |

Use only 00, 01, 09 for shifts

Use OF with n = 0 for final stop

| 1V n | Right Shift of AQ | 3 Orders |
|---|---|---|

If n = 0, the machine will stop. If not, repeat n times the operation which replaces the contents

$a_0, a_1, a_2, \ldots, a_{38}, a_{39}$ of A

and

$q_0, q_1, q_2, \ldots, q_{38}, q_{39}$ of Q

by

$a_0, a_0, a_1, \ldots, a_{37}, a_{38}$

and

$q_0, a_{39}, q_1, \ldots, q_{37}, q_{38}$.

The number n will be interpreted modulo 64.

Variants

| | |
|---|---|
| 0, 2, 4, 6 | Order as described above. |
| 1, 3, 5, 7 | Clear A and then execute as described. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9, S, J, L | Clear A, insert 1/2 in A, and then execute order as described above. |

Use only 10, 11, 19

## 2V n Unconditional Control Transfer 12 Orders

Bring the next order pair from memory location n and choose the left or right hand order of this pair, stopping beforehand or not, depending upon V.

### Variants

0 — Stop. The first order after starting with the START switch will be the right hand order at memory location n. The stop can be ignored by setting a panel switch.

2 — Transfer control to right hand order at memory location.

4 — Same as 0 except take left hand order.

6 — Same as 2 except take left hand order.

1, 3, 5, 7 — Same as 0, 2, 4, 6 except clear A first.

8, K, N, F — Illiac will hang up. Avoid these.

9, S, J, L — Same as 1, 3, 5, 7 except also make A = 1/2 after clearing.

USE 20, 21, 22, 23, 24, 25, 26, 27, 29, 2S, 2J, 2L.

STARTING AFTER A STOP. When the ILLIAC has been stopped by one of the control transfer stop orders, it is usually started again by moving the black switch to START, from which position the switch automatically returns to OBEY.

The Illiac can also be started again by moving the white switch through EXECUTE to FETCH and then back to RUN. If this is done, the control transfer order which stopped the Illiac will be ignored. The normal sequencing will then follow unless the stop order is a right hand order

transferring control to the right hand side of a word. In this case, the order first obeyed after starting will be the right hand instead of the left hand order of the order pair brought out.

For example consider the following orders:

| | | |
|---|---|---|
| p | L5 | F |
| | 20 | p+2 |
| p+1 | 40 | 1F |
| | 24 | p |
| p+2 | 7J | 2F |
| | L4 | 3F |

If we stop with the 20 order, the black switch will start with L4 3F and the white switch with 24 p. If we stop with the 24 order, the black switch will start with L5 F and the white switch with 7J 2F.

## 3V n Conditional Control Transfer 4 Orders

If $A \geqslant 0$, bring the next order pair from memory location n and choose the left or right hand order of this pair, stopping beforehand or not, depending upon V. If $A < 0$, go on to the next order.

### Variants

| | |
|---|---|
| 0, 2, 4, 6 | If $A \geq 0$, do the same operation as for the corresponding 2V order. If $A < 0$, go on to the next order. |
| 1, 3, 5, 7 | Identical with corresponding 2V orders. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9, S, J, L | Identical with corresponding 2V orders. |

Use only 30, 32, 34, 36.

### Starting After A Stop

The discussion given with the 2V orders applies here to the corresponding 3V orders.

4V n　　Store　　9 Orders

Copy into memory location n all of the contents of A, the contents corresponding to the address of a left hand order, or the contents corresponding to the address of a right hand order, depending upon V.

### Variants

| | |
|---|---|
| 0, 4 | Replace N(n) by A. |
| 1, 5 | Replace N(n) and A by 0. |
| 2 | Replace address digits of right hand order at memory location n by the corresponding digits of A. |
| 3 | Same as 2 except clear A first. |
| 6 | Same as 2 except take left hand order. |
| 7 | Same as 6 except clear A first. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9, J | Replace N(n) and A by 1/2. |
| S | Replace A by 1/2 and address digits of right hand order at memory location n by o. |
| L | Same as S except take left hand order. |

Use only 40, 41, 42, 43, 46, 47, 49. 4S, 4L.

| | | |
|---|---|---|
| 5V n | Memory to Q | 3 Orders |

Transfer N(n) to Q.

Variants

| | |
|---|---|
| 0, 2, 4, 6 | Transfer N(n) to Q. |
| 1, 3, 5, 7 | Clear A and transfer N(n) to Q. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9, S, J, L | Put 1/2 in A and transfer N(n) to Q. |

Use only 50, 51, 59.

6V n Divide 3 Orders

Divide AQ by N(n), placing the rounded quotient in Q (the least significant digit being 1 for the roundoff) and leaving a residue in A. If $|A| > |N(n)|$ the Illiac will stop after dividing. If $|A| = |N(n)|$ and if $A \geq 0$, the Illiac will stop after dividing; if $A < 0$, the Illiac will not stop after dividing.

## Variants

| | |
|---|---|
| 6 | As described above. |
| 7 | Make A = 0, Then proceed as above. |
| L, S | Make A = 1/2. Then proceed as above. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 0, 1, 2, 3, 4, 5,<br>9, J | These give incorrect results or results which are correct only under certain conditions. Avoid them. |
| | Use only 66, 67, 6L. |

7V n Multiply 12 Orders

Put Q x P(n) + $2^{-39}$ A into AQ the least significant 39 digits being in Q with $q_0 = 0$.

## Variants

| | |
|---|---|
| 0 | P(n) = -N(n) |
| 1 | P(n) = -N(n); A = 0 "UNROUNDED NEGATIVE MULTIPLY" |
| 2 | P(n) = - $\lvert$N(n)$\rvert$ |
| 3 | P(n) = - $\lvert$N(n)$\rvert$ ; A = 0 |
| 4 | P(n) = N(n). "HOLD MULTIPLY" |
| 5 | P(n) = N(n); A = 0 "UNROUNDED MULTIPLY" |
| 6 | P(n) = $\lvert$N(n)$\rvert$ |
| 7 | P(n) = $\lvert$N(n)$\rvert$ ; A = 0 |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9 | P(n) = - N(n); A = 1/2. "ROUNDED NEGATIVE MULTIPLY" |
| S | P(n) = - $\lvert$N(n)$\rvert$ ; A = 1/2 |
| J | P(n) = N(n); A = 1/2 "ROUNDED MULTIPLY" |
| L | P(n) = $\lvert$N(n)$\rvert$ ; A = 1/2 |

Use only 70, 71, 72, 73, 74, 75, 76. 77. 79, 7S, 7J, 7L.

8V n Input-Output 9 Orders

Transfer words between A and the input tape, output punch, or magnetic drum.

The address n must be a multiple of 4 for the tape and punch orders and is interpreted modulo 64. The address must be 11 for drum orders.

Variants

0 Shift AQ 4 places left as in the 00 order and replace $a_{36}$, $a_{37}$, $a_{38}$, $a_{39}$ by the binary digits corresponding to the sexadecimal character being read. This is done n/4 times.

1, 9 Clear A and then do as in 80 order.

2 Punch the digits $a_0$, $a_1$, $a_2$, $a_3$ as one sexadecimal character and shift AQ 4 places left as in the 00 order. This is done n/4 times.

3 Clear A and do as in 82 order.

5 This is a 40-digit order of the form 85 11 TV p. We distinguish two cases.

(1) T is not 0, 1, 8, 9. In this case, A and Q are shifted left 11 places as in the 00 order and the word at drum location p is placed in A. Then the T order is executed using address p. Complete freedom is not available in drum addresses because p may interfere with V.

(2) T is 0, 1, 8, 9. In this case after the word at drum location p is placed in A the right hand order is skipped. This permits use of any drum address for p.

6 This is a 40-digit order of the form 86 11 TV p. We again have two cases:

(1) T is not 0, 1, 8, 9. In this case A is transferred to drum location p, and A and Q are shifted 11 places left as in the 00 order. Then the T order is executed using address p. Complete freedom is not available in drum addresses because p may interfere with V.

(2) T is 0, 1, 8, 9. In this case the right hand order is skipped after doing the left hand order as in case (1). This permits use of any drum address p.

7 Same as 86 except clear A first.

S Put 1/2 in A and do as in 82 order.

L Put 1/2 in A and do as in 86 order.

8, K, N, F Illiac will hang up. Avoid these.

4, J These are not useful. Avoid them.

Use 80, 81, 82, 83, 85, 86, 87, 8S, 8L.

9V n Special Input-Output 1 Order

Variants

2 Letter output. Punch on the tape a character depending upon the address digits n. Three quantities are defined by the 10 binary address digits:

(1) The leftmost 4 digits define the usual 4 digit positions in the output tape.

(2) The rightmost digit defines the 5th hole in the output tape.

(3) The rightmost 6 digits determine the number b of times that the above-defined character is punched and also the number of right shifts executed. The number of characters punched will be found by dividing the number in the rightmost 6 digits by 4 and rounding up to the next integer.

The address n may always be found from the following formula:

$$n = 64a + 4b + c - 2$$

where a is the character punched, a = 0, 1, 2, ... , J, F, L

b is the number of characters punched, $1 \leq b \leq 16$

c determines the fifth hole, c = 0, 1.

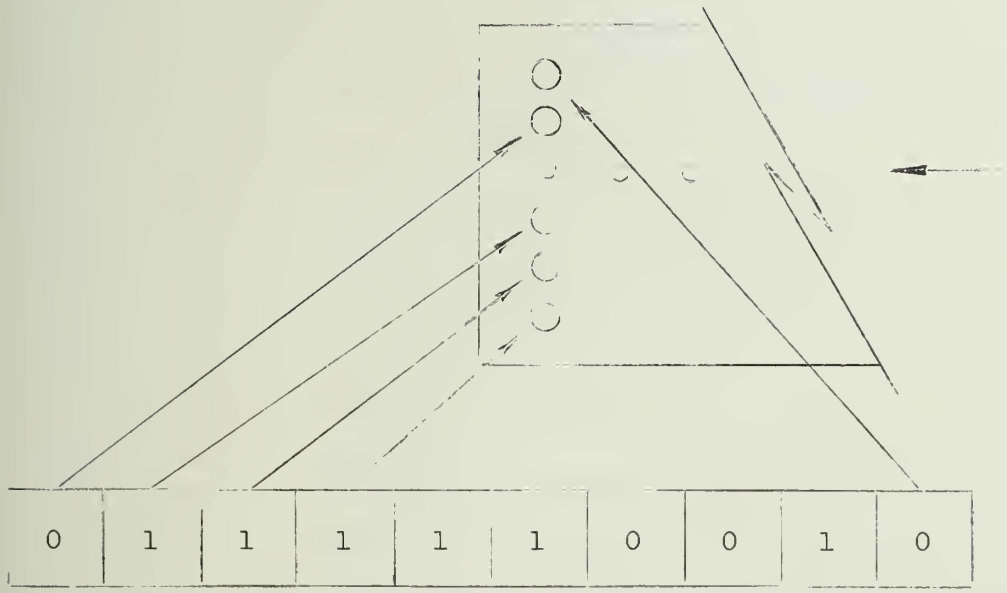
The number of right shifts executed is $4b + c - 2$.

Example. Punch the character 7 thirteen times.

$$n = 64 \times 7 + 4 \times 13 + 0 - 2 = 498$$

It will be found that the last 6 digits contain the number 50 which when divided by 4 and rounded up gives 13. There will be 50 right shifts.

Figure 3.2 shows the relationship between the tape holes and the address digits of the 92 order.



The address shown (decimal 498) will print the character 7 thirteen times.

FIGURE 3.2. ADDRESS DIGITS OF 92 ORDER

KV n Increment Add from Q 12 Orders

These orders are identical with the FV orders with Q instead of N(n).

## Variants

| | |
|---|---|
| 0 | Add $-Q - 2^{-39}$ to A. (See footnote). |
| 1 | Put $-Q - 2^{-39}$ in A. |
| 2 | Same as 0 if $Q \geq 0$; same as 4 if $Q < 0$. |
| 3 | Same as 1 if $Q \geq 0$; same as 5 if $Q < 0$. |
| 4 | Add $Q + 2^{-39}$ to A. |
| 5 | Put $Q + 2^{-39}$ to A. |
| 6 | Same as 4 if $Q \geq 0$; same as 0 if $Q < 0$. |
| 7 | Same as 5 if $Q \geq 0$; same as 1 if $Q < 0$. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9 | Put $-Q - 2^{-39} + 1/2$ in A. |
| S | Same as 9 if $Q \geq 0$; same as J if $Q < 0$. |
| J | Put $Q + 2^{-39} + 1/2$ in A. |
| L | Same as J if $Q \geq 0$; same as 9 if $Q < 0$. |

Use K0, K1, K2, K3, K4, K5, K6, K7, K9, KS, KJ, KL.

NOTE: $-Q - 2^{-39}$ is the digitwise complement of Q.

SV n    Add from Q    12 Orders

Q is added, subtracted, etc., to A. These orders are identical with the LV orders with Q instead of N(n).

## Variants

| | |
|---|---|
| 0 | Subtract Q from A. |
| 1 | Put -Q in A. |
| 2 | Subtract $\|Q\|$ from A. |
| 3 | Put $-\|Q\|$ in A. |
| 4 | Add Q to A. |
| 5 | Put Q in A. |
| 6 | Add $\|Q\|$ to A. |
| 7 | Put $\|Q\|$ in A. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9 | Put 1/2 - Q in A. |
| S | Put $1/2 - \|Q\|$ in A. |
| J | Put 1/2 + Q in A. |
| L | Put $1/2 + \|Q\|$ in A. |

Use S0, S1, S2, S3, S4, S5, S6, S7, S9, SS, SJ, SL

JV n Extract 3Orders

If two corresponding digits of N(n) and Q are both 1's, put 1 in that place of Q. Otherwise put 0. This order gives the logical product of N(n) and Q.

## Variants

| | |
|---|---|
| 0, 2, 4, 6 | As described above. |
| 1, 3, 5, 7 | Clear A and do J0 order. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9, S, J, L | Put 1/2 in A and do J0 order. |

Use only J0, J1, J9.

FV n Increment Add 12 Orders

## Variants

| | |
|---|---|
| 0 | Add $-N(n) - 2^{-39}$ to A, (See footnote). |
| 1 | Put $-N(n) - 2^{-39}$ in A. |
| 2 | Same as 0 if $N(n) \geq 0$; same as 4 if $N(n) < 0$. |
| 3 | Same as 1 if $N(n) \geq 0$; same as 5 if $N(n) < 0$. |
| 4 | Add $N(n) + 2^{-39}$ to A. |
| 5 | Put $N(n) + 2^{-39}$ in A. |
| 6 | Same as 4 if $N(n) \geq 0$; same as 0 if $N(n) < 0$. |
| 7 | Same as 5 if $N(n) \geq 0$; same as 1 if $N(n) < 0$. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9 | Put $-N(n) - 2^{-39} + 1/2$ in A. |
| S | Same as 9 if $N(n) \geq 0$; same as J if $N(n) < 0$. |
| J | Put $N(n) + 2^{-39} + 1/2$ in A. |
| L | Same as J if $N(n) \geq 0$; same as 9 if $N(n) < 0$. |

Use F0, F1, F2, F3, F4, F5, F6, F7, F9, FS, FJ, FL.

NOTE: $-N(n) - 2^{-39}$ is the digitwise complement of N(n).

LV n Add 12 Orders

N(n) is added, subtracted, etc., to A.

## Variants

| | |
|---|---|
| 0 | Subtract N(n) from A. |
| 1 | Put -N(n) in A. |
| 2 | Subtract $|N(n)|$ from A. |
| 3 | Put - $|N(n)|$ in A. |
| 4 | Add N(n) to A. |
| 5 | Put N(n) in A. |
| 6 | Add $|N(n)|$ to A. |
| 7 | Put $|N(n)|$ in A. |
| 8, K, N, F | Illiac will hang up. Avoid these. |
| 9 | Put 1/2 - N(n) in A. |
| S | Put 1/2 - $|N(n)|$ in A. |
| J | Put 1/2 + N(n) in A. |
| L | Put 1/2 + $|N(n)|$ in A. |

Use $\underline{\text{L0}}$, $\underline{\text{L1}}$, $\underline{\text{L2}}$, $\underline{\text{L3}}$, $\underline{\text{L4}}$, $\underline{\text{L5}}$, $\underline{\text{L6}}$, $\underline{\text{L7}}$, L9, LS, LJ, LL.

## ABBREVIATED ORDER LIST

| ORDER | DESCRIPTION |
|---|---|
| 00 n | Shift AQ left n places, $1 \leq n \leq 63$. |
| OF 0 | Final stop. |
| 10 n | Shift AQ right n places, $1 \leq n \leq 63$. |
| 19 n | Make A = 1/2 and shift AQ right n places so that AQ contains $2^{-n-1}$, $1 \leq n \leq 63$. |
| 20 n | Stop. The first order after START will be the right hand order at location n. The stop can be ignored by setting the panel switch to IGNORE. |
| 22 n | Transfer control to the right hand order at location n. |
| 24 n | Same as 20 except take left hand order. |
| 26 n | Same as 22 except take left hand order. |
| 30 n | |
| 32 n | If $A \geq 0$ do as in the corresponding 2V order. |
| 34 n | If $A < 0$ go on to the next order. |
| 36 n | |
| 40 n | Replace N(n) by A. Do not change A. |
| 41 n | Replace N(n) and A by zero. |
| 42 n | Replace address digits of the right hand order at location n by the corresponding digits of A. Do not change A. |
| 46 n | Replace address digits of the left hand order at location n by the corresponding digits of A. Do not change A. |
| 50 n | Replace Q by N(n). |
| 66 n | Divide $A + 2^{-39} (Q + q_0)$ by N(n). That is, form AQ/N(n). |

| ORDER | DESCRIPTION |
|---|---|
| 71 n | Multiply -N(n) by Q, putting result in AQ. |
| 74 n | Put N(n) Q + $2^{-39}$A into AQ. |
| 75 n | Multiply_N(n) by Q, putting result in AQ. |
| 79 n | Put the rounded product -N(n) Q into A. |
| 7J n | Put the rounded product N(n) Q into A. |
| 81 n | Clear A and input n/4 sexadecimal characters from the tape. |
| 82 n | Punch n/4 sexadecimal characters on the tape. |
| 85 n | Replace A by the word at drum location n. See Page 17. |
| 86 n | Replace the word at drum location n by A. See Page 17. |
| 92 131 | Punch one carriage return and line feed character. |
| 92 642 | Punch one + sign (or K). |
| 92 706 | Punch one - sign (or S). |
| 92 963 | Punch one space character. |
| 92 515 | Delay |
| K5 n | Put Q + $2^{-39}$ in A. |
| S0 n | Subtract Q from A. |
| S1 n | Put -Q in A. |
| S2 n | Subtract $\lvert Q \rvert$ from A. |
| S3 n | Put - $\lvert Q \rvert$ in A. |
| S4 n | Add Q to A. |
| S5 n | Put Q in A. |
| S6 n | Add $\lvert Q \rvert$ to A. |
| S7 n | Put $\lvert Q \rvert$ in A. |
| J0 n | Replace Q by the logical product of Q and N(n). |

| ORDER | DESCRIPTION |
|---|---|
| F4 n | Add $N(n) + 2^{-39}$ to A. |
| F5 n | Put $N(n) + 2^{-39}$ in A. |
| L0 n | Subtract N(n) from A. |
| L1 n | Put -N(n) in A. |
| L2 n | Subtract $\lvert N(n) \rvert$ from A. |
| L3 n | Put - $\lvert N(n) \rvert$ in A. |
| L4 n | Add N(n) to A. |
| L5 n | Put N(n) in A. |
| L6 n | Add $\lvert N(n) \rvert$ to A. |
| L7 n | Put $\lvert N(n) \rvert$ in A. |